

ASC Portal Users Guide

Ruxandra Bondarescu * Gregory Daues †

July 26, 2002

Abstract

The Astrophysics Simulation Collaboratory (ASC) project seeks to provide a web-based problem solving framework for the astrophysics community to harness computational Grids. In addition to the generic environment suitable for any application the ASC portal supplies a specialized framework for the Cactus Computational toolkit.

The Cactus computational toolkit provides a widely used software package for solving partial differential equations in 3D including Einsteins general relativistic wave equations. The modular design of the Cactus framework permits a diverse group of researchers to develop add-on software modules, termed “thorns”, to integrate additional physics or numerical solvers into the Cactus framework.

The ASC Portal is a Web-based application server that provides access to Grid services and resources such as high-performance computers, code repositories, information servers, etc, and maintains a database documenting the history of each user’s interaction with the Grid. This document navigates the user from the starting point though the full range of interfaces that enable grid computing.

Contents

1	Getting started with the ASC Portal	2
2	Grid	3
2.1	Credentials	3
2.2	Resources	4
2.3	Services	4

*ruxandra@ncsa.uiuc.edu

†daues@ncsa.uiuc.edu

3	Cactus	4
3.1	Demo	4
3.2	Configuration	6
3.3	Simulations	10
3.4	Parameters	11
4	Utilities	12
4.1	FTP	12
4.2	GRAM	13
4.3	LDAP	16
5	Preferences	17
6	Visualization	17
7	Conclusion	19

1 Getting started with the ASC Portal

In order to perform grid operations within a portal session, one retrieves a grid credential from the ASC organizational MyProxy server. In order to load a credential onto this server it is necessary to have MyProxy client software installed on the same system on which your grid certificate/key pair are stored. A grid certificate is used to have a single sign-on access to all of the grid-enabled machines on which you have accounts.

The Myproxy package provides a secure method for portal users to access resources with a limited proxy using the Globus Security Infrastructure (GSI). If the web server is broken into, only the time limited delegated credentials are compromised. The proxy gives you single sign-on capability. While your proxy is active, you can log into any Grid resource without re-entering your pass phrase. A proxy is loaded onto the ASC organizational MyProxy server by running the myproxy-init command:

```
myproxy-init -s myproxy.ascportal.org
```

or

```
myproxy-init -l username -s myproxy.ascportal.org
```

This command will generate a grid proxy credential, and store that credential on the MyProxy server. Within an ASC Portal user session one retrieves this credential for authenticating to Grid sites, enabling the user to perform grid operations such as checking out source code, staging files and submitting jobs. The myproxy-init command will require you to enter the password you used to generate the private key of your default grid credential, and then to enter a new password for retrieving your grid credential at a later time. It is highly recommended you choose a password other than the one you used to generate the private key of your grid credential. The username is used when you create your ASC Portal credential and it is set by default as your username from the machine you are running the myproxy-init command.

The first time that you login to the portal you will need to use your default username and a corresponding proxy password. After this initial login, credential definitions can be adjusted to make use of a different username.

Upon logging in to the portal you see the current status, consisting of a description of the present user session and a list of your current user credentials. In the Main Menu on the left-hand side are the following options: “Home”, “Grid”, “Cactus”, “Utilities”, “Preferences”, “Visualization” and “Help”. In the subsequent sections of this document we present the full description of these portal components.

2 Grid

Grids are world wide collections of high performance resources. Grids bring together geographically and organizationally dispersed computational resources such as widely distributed storage systems, computing platforms, instrumentation, and visualization engines. The grid resources available from the portal are machines that have Globus installed.

2.1 Credentials

A Grid user will typically possess certificates signed by various virtual organizations. Interfaces underneath Grid->Credentials manage multiple certificates for portal users. When activating a given credential the username and password must be the same as those entered to protect the proxy when running the myproxy-init command. Different credential definitions must be assigned different usernames. When you log into the portal for the first time you use your default portal username. In this section you can create credentials with different usernames. Activation of an alternative credential will require a corresponding execution of :

```
myproxy-init -l alternative_username -s myproxy.ascportal.org
```

This enables you to subsequently log in to the portal with your portal username and any of your active proxies passwords.

2.2 Resources

Information about the resources within the user's "Virtual Machine Room" is displayed on these pages. This information about machines and their properties is pulled from the portal database records. Utilities that transfer information from organizational LDAP servers to the database are still being written. Consequently many records are incomplete, but useful elements about popular machines have been entered (e.g., batch queues the NCSA Origin, Potsdam Origin, WashU Origin, Platinum IA32 Linux cluster).

2.3 Services

Portal administrators are responsible for configuring the list of available services. This list is shown on the main Grid-> Services page. The interface provided lets the user add or remove individual services to the user's profile. An authentication test is performed before entering the service into the profile. The services list shows the certificate that succeeded in authenticating.

3 Cactus

Cactus is an application middleware for parallel computation. The name of Cactus comes the design of a central core (or "flesh") which connects to application modules (or "thorns") through an extensible interface. The flesh provides the interface and structure for Cactus, while the thorns provide the applications and the computational infrastructure.

3.1 Demo

This section enables users to run demo cactus simulations, i.e., the user is not required to have their own cactus executables or parameter files. Instead the executable is staged from a single repository maintained by portal administrators. The gatekeeper on the target host is directed to the address of the executable on the repository (for example, `modi4.ncsa.uiuc.edu/jobmanager` is told to execute `http://nexus.ncsa.uiuc.edu:8080/cactus/cactus_wave`). The parameter file is transferred using GridFTP from the same repository to the target host.

This repository presently holds WaveToy executables compiled for SGI and Linux platforms. In the demo section one launches the cactus simulation via GRAM and the demo attributes are recorded in the portal database. The user can see a history of submitted cactus demos on the local demos page. Finer details of the job submission can be viewed in the Utilities->GRAM section.

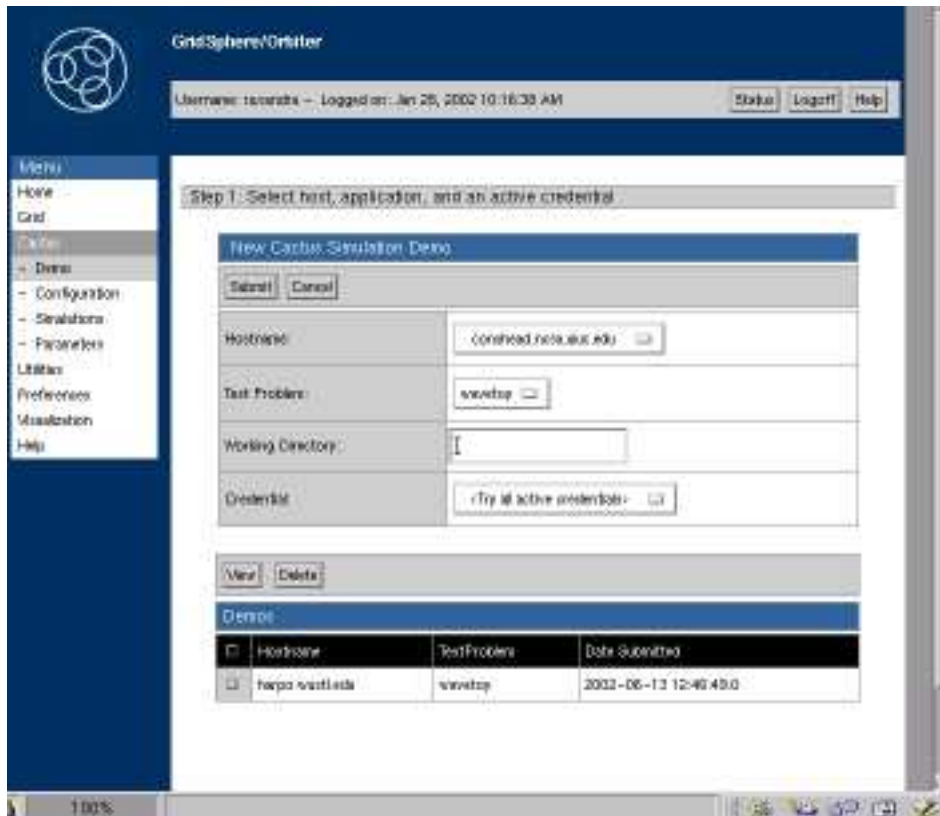


Figure 1: The interface for starting a Cactus demo simulation

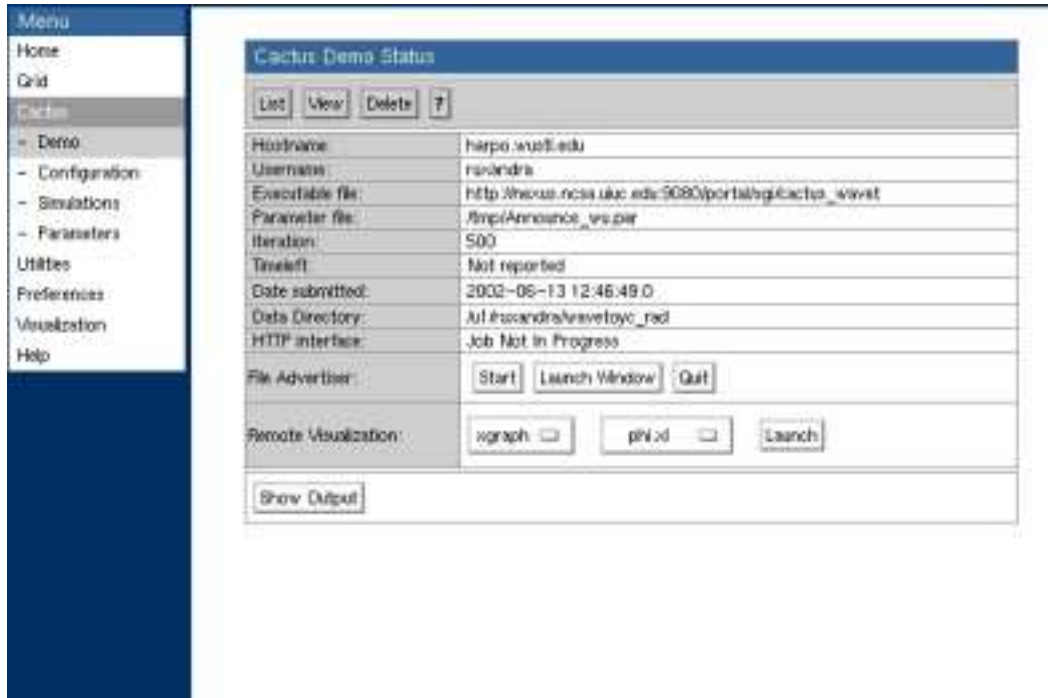


Figure 2: The Cactus-demo view page

Once the demo simulation is started the job output can be viewed in an interactive Java Applet. The present iteration number, time per iteration and a list with the generated data files are displayed in plain html by reloading the view page. While the simulation is in progress an HTTP interface providing further information about the running Cactus simulation (included Cactus Thorns, parameter file, time to completion, downloadable output files, etc.). From the HTTP interface one can choose to cancel the simulation. If the demo simulation is completed the user can launch the File Advertiser to restart the same HTTP interface. A listing of output files in the data directory is gathered via GridFTP and displayed to the users. One can launch a simple visualization tool on the remote machine (for now xgraph and gnuplot) to plot the generated data files.

In order to launch any visualization tools through the portal the user is required to have an X environment. Additionally, prior to launching a visualisation tool one needs to run `xhost +<remote_host>` on the local client machine.

After the job is done the HTTP sever can be restarted using the Cactus File Advertiser. If the file advertiser fails to launch try pressing “Quit” to clear the file advertiser processes on the remote host and the portal database entries. Pressing “Quit” before logging out of the portal is recommended.

3.2 Configuration

This section will allow the user to perform standard cvs checkouts (no grid security, yet) of the cactus code. The portal runs cvs checkout as a GRAM job. The user can checkout Cactus Flesh, entire arrangements or individual thorns.

The portal maintains a permanent list with the Cactus setups built by users in this interface. This list is displayed on the main configurations page. Each configuration contains the full history of the cvs checkouts performed in the portal for the corresponding hostname and top directory.

The view-page provides detailed information about the selected Cactus configuration (i.e., the job(s) status, the top directory, the list of arrangements and thorns checked out in the portal, etc) The user is also able to “Modify” a configuration (i.e., add arrangements and thorns).

In order to checkout arrangements and thorns you must have already executed the checkout of “Cactus Flesh”. The checkout of arrangements and thorns is performed in `top_directory/Cactus/arrangements`. The arrangements and Cactus directories are created when you check out the Cactus Flesh.

The Top Directory is the relative path were you want to checkout cactus. For example if you have a directory called “work” in your home directory the top directory will be “work”. The user can also write the full path if desired.

The “Duplicate”, “Login Setup” and “Repositories” choices lead to detailed interfaces which we discuss bellow.

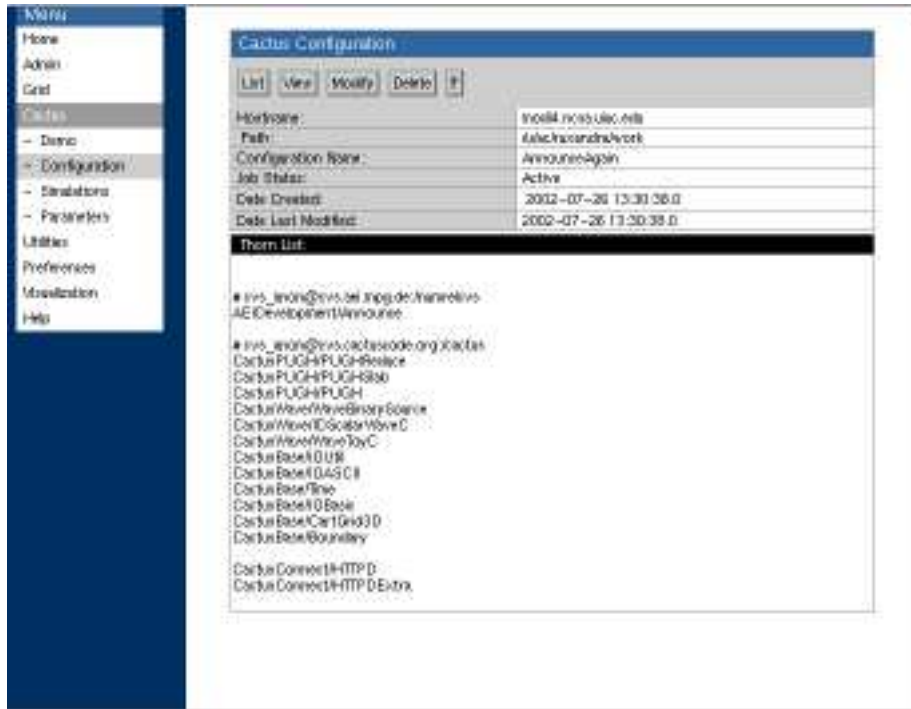


Figure 3: Cactus Configuration built within the portal

- Duplicate

One can duplicate a configuration that resides on one machine on a second resource. The method that we use to accomplish this is to perform the same sequence of cvs checkouts on the remote machine (we keep track of the previous cvs checkouts using a portal thorn list).

To duplicate a configuration you have to specify:

- Hostname

The machine on which you want to duplicate the configuration. Note, that by default you will get the version of the code that is presently in CVS. The checkout of the Cactus flesh is done by default from: `cvs.cactuscode.org:/cactus`. Also, the portal does not keep track of the cvs checkout commands done outside the portal interface.

- Configuration name

Presently this is just a label that appears only in the portal. We recommend changing the configuration name to avoid confusion.

- Top Directory

The Top Directory is the relative path were you want to checkout Cactus. For example if you have a directory called “work” in your home directory the top directory will be “work”. The user can also write the full path if desired.

Once the duplication process is started one can check its status in the configuration-view page. The process is declared completed when all the cvs checkout jobs are done.

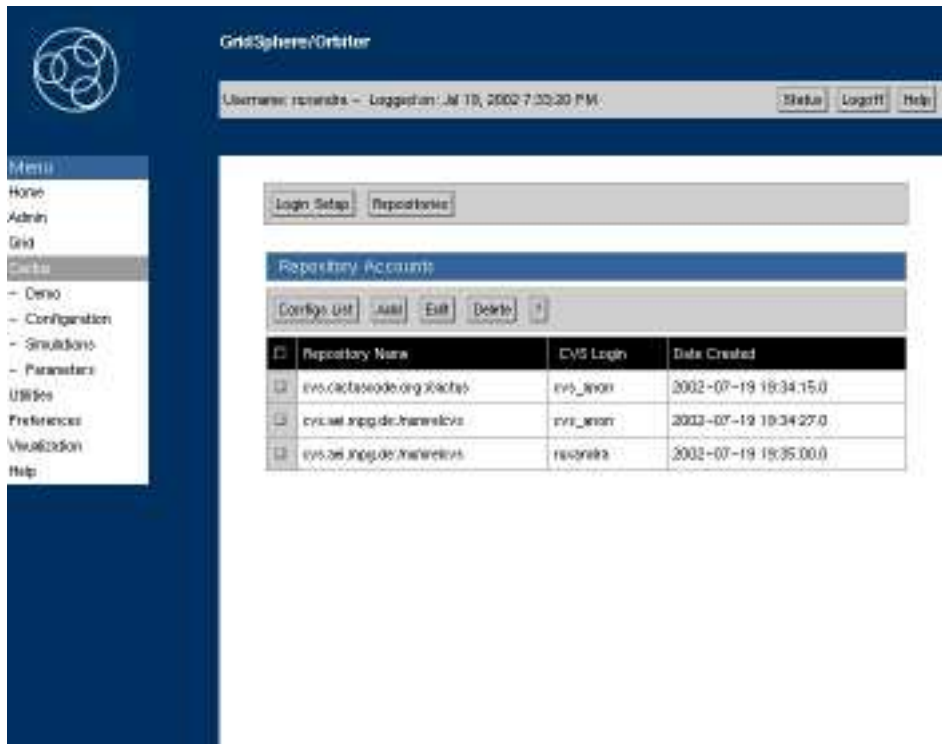


Figure 4: Portal records of CVS repository accounts

- CVS Login Setup

The portal keeps track of cvs logins and passwords, keeping the user from having to reenter these each time. The user simply enters them once into the setup interface before initiating any cvs checkouts through the portal.

- Add

Creates a new repository account in the portal database. Please note that even if you want to perform anonymous checkouts you should create an entry (cvs_anon, password: anon) for each repository.

- Edit

Enables the user to modify the portal record of his/her repository account.

- Delete

Removes the selected repository account(s) from the portal database.

The notion of a “Repository Account” added in this interface is just a portal database record and has no influence on actual cvs servers and cvs user accounts.

- CVS Repositories

The user can configure their personal cvs repositories and their contents for use in the portal. Portal administrators perform this task for main group cvs repositories.

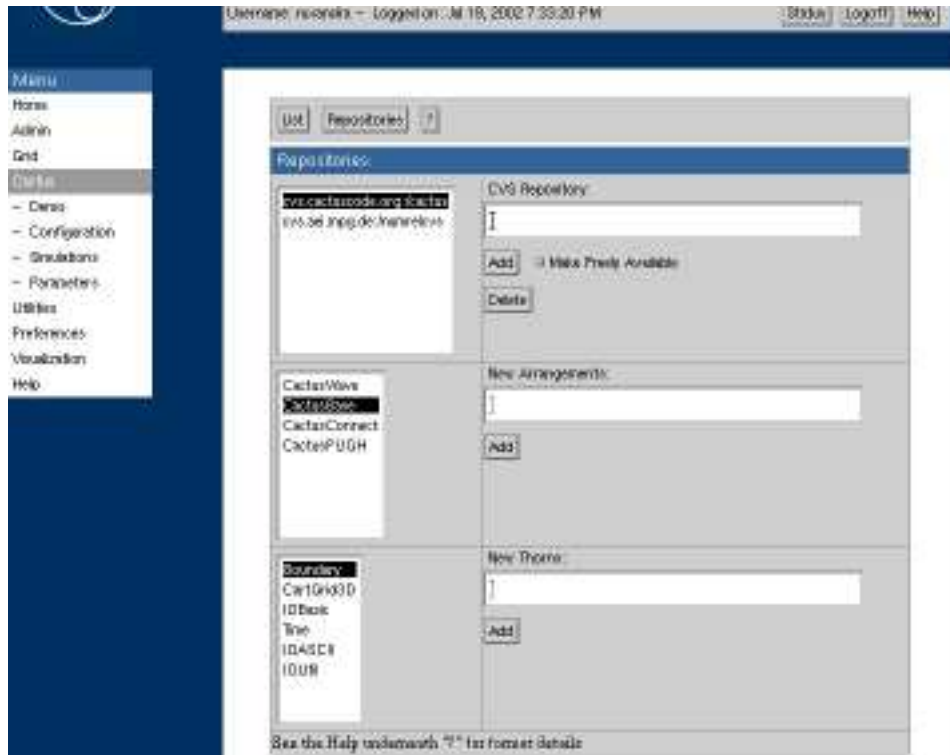


Figure 5: CVS repositories contents stored in the portal

– Add

In the repository section of the interface this button enables the user to add a repository named in the corresponding text field. One can make a repository freely available to other group members by checking the adjacent box. Note that once a repository is made freely available only a portal admin can remove it.

In the arrangements section the user can add multiple arrangements at a time, separating the values by a comma within the list. For example, in the `cvs.cactuscode.org/cactus` repository one can enter: `CactusBase,CactusConnect,CactusWave`. The repository has to be previously highlighted.

One can add multiple thorns to a single highlighted arrangement by entering a comma-separated list in the text field. For example if one selects the `cvs.cactuscode.org/cactus` repository and `CactusConnect` arrangement he/she can enter the ThornList:

`HTTPD,HTTPDExtra`

– Delete

Removes the selected repository from the portal database. Only personal repositories can be removed by portal users. Portal administrators may delete the globally available cvs servers.

Hostname	Username	Executable	Parameters file	Date Submitted
nrc01.nrc.ca.us	nrc01nrc	cactus_ave1	Announce.par	2002-05-16 18:18
nrc02.nrc.ca.us	nrc02nrc	cactus_ave1	Announce.par	2002-05-16 12:16
origin.aei-potsdam.inpg.de	inhal	cactus_ave1	Announce.par	2002-04-23 21:50
origin.aei-potsdam.inpg.de	ryol	cactus_jit6	sergio_ji2s10.75_2.6.par	2002-04-22 11:58
origin.aei-potsdam.inpg.de	ryol	cactus_jit6	sergio_ji2s10.75_2.6.par	2002-04-22 11:33
harpo.vuwi1.edu	tradio	cactus_gab	Demo41Output.par	2002-04-22 07:38
veon04.aei-potsdam.inpg.de	tradio	cactus_gab	Demo41Output.par	2002-04-22 06:57
veon04.aei-potsdam.inpg.de	tradio	cactus_gab	Demo41Output.par	2002-04-22 06:56
origin.aei-potsdam.inpg.de	ryol	cactus_jit6	sergio_ji2s10.75_2.6.par	2002-04-22 06:52
harpo.vuwi1.edu	allen	cactus_ave1_single	Demo41Output.par	2002-04-22 05:07
origin.aei-potsdam.inpg.de	ryol	cactus_jit6	sergio_ji2s10.75_2.6.par	2002-04-22 04:38
origin.aei-potsdam.inpg.de	ryol	cactus_jit6	sergio_ji2s10.75_2.6.par	2002-04-22 03:53
harpo.vuwi1.edu	allen	cactus_ave1_single	Demo41Output.par	2002-04-21 10:44
harpo.vuwi1.edu	allen	cactus_ave1_single	Demo41Output.par	2002-04-21 10:43
origin.aei-potsdam.inpg.de	ryol	cactus_jit6	sergio_ji2s10.75_2.6.par	2002-04-19 11:02
origin.aei-potsdam.inpg.de	ryol	cactus_jit6	sergio_ji2s10.75_2.6.par	2002-04-19 10:58
origin.aei-potsdam.inpg.de	ryol	cactus_jit6	sergio_ji2s10.75_2.6.par	2002-04-19 10:24

Figure 7: Cactus Simulations Archive Page

A Cactus simulation can be set up to communicate to the portal via the XMLRPC mechanism whether it is launch inside the portal or on the command line. It communicates information designated at run time such as the HTTP port where the simulation is running, the directory where the data files are stored, the current iteration number, and the time left to completion. Cactus simulations can also invoke a deannounce method to remove themselves from the portal database.

The Archive page of the simulations section displays internally and externally submitted simulations as mentioned above. The external announced simulation are divided into two sections: the simulations that announced to the portal in the last 24 hours, and previously announced simulations. The display of the present day's simulations is a helpful collaborative tool, as other group members can readily monitor active simulations.

3.4 Parameters

This utility is really a general text file importer/exporter; A complete listing of the remote file system similar to the one in the Utilities->FTP interface is displayed.

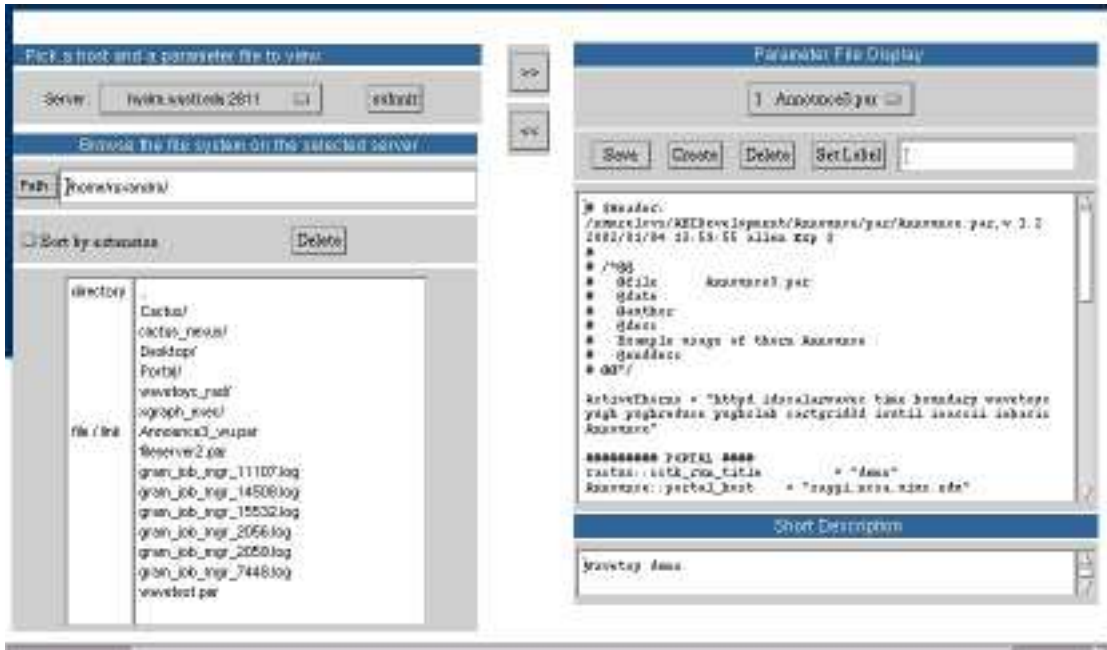


Figure 8: The parameter file interface

The portal keeps a permanent collection of parameter files for each user. One can import a text file from a grid resource into this collection on the portal, where it can be edited and then exported to the same or to a different grid resource. There are numerous functions implemented to allow the user to manipulate the file in the local repository. One can modify the file in the portal database, rename it, create a new local file, delete entries from the database, attach a short description of a parameter file and then export it to the desired location on a remote host. The user's local collection of parameter files is maintained across portal sessions.

4 Utilities

4.1 FTP

This Utility performs grid enabled file transfers between resources that use GSI authentication. It also allows the user to manage the filespace on remote resources.

The interface can display the file listing of any readable directory on the remote machine. Directories and files are clearly marked and listed separately from one another. A full unix file description (complete file name, file size, date last modified) is displayed when the file/directory is selected. The file listing can ordered by the filename extensions. The user can navigate easily through the file space with mouse clicks. Standard unix file space management is also provided. One can create new directories, delete files and rename existing files. File transfers can be done in several ways: individual file transfers, entire directory transfers, and multiple file tranfers. In each of these cases

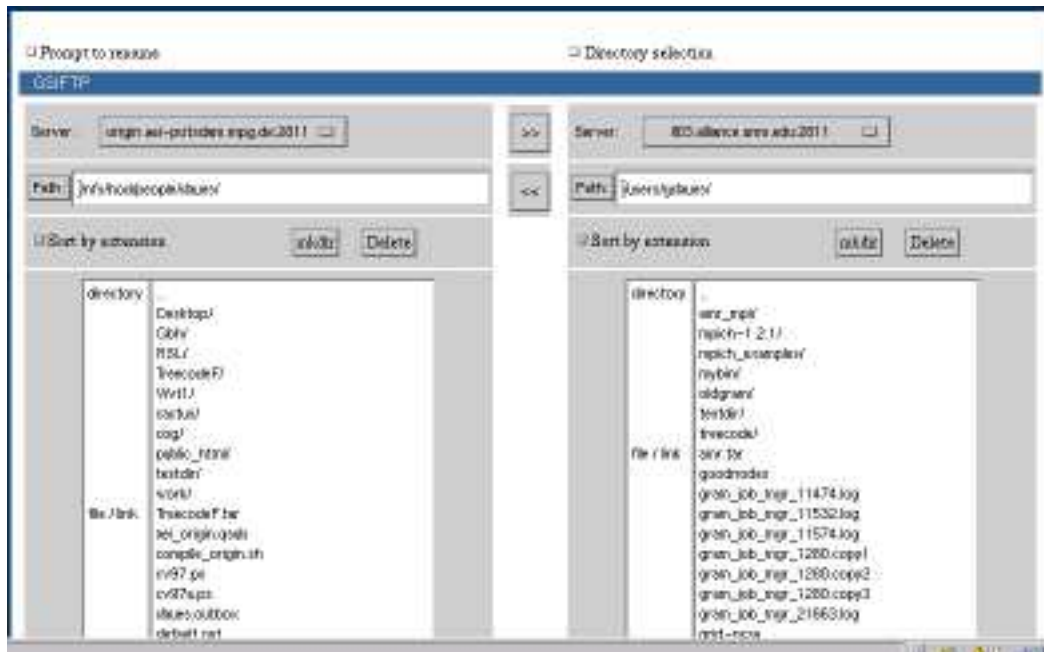


Figure 9: The FTP interface

the user can choose to rename the selected files or directories.

Our code that enables the file transfers is based on the Java CoG. It uses the GridFTPClient class. New instances of this class are created each time the ftp interface page is loaded in the browser window. In the authentication process the portal loops through the user's active credentials to determine which type of certificate is accepted by the remote host. While file listings on a given machine succeed for multiple certificates (assuming proper grid-mapfile entries), for successful file transfers the certificate used to authenticate to the two grid machines must be signed by the same authority. Also, the GridFTP daemons running on remote supercomputers must be configured to support third party file transfers for this software to work.

4.2 GRAM

The GRAM section of the portal enables the user to submit a job to a remote resource through the web interface. The Globus Resource Allocation Manager (GRAM) is the lowest level of Globus resource management architecture. GRAM allows the user to run jobs remotely, providing an API for submitting, monitoring, and terminating your job.

This section contains a thorough interface for preparation of a GRAM job submission. Through the portal interface one can submit jobs to interactive or batch systems. A complete list of the batch queues on the remote supercomputer is provided. In the near future, this list will provide load and scheduling information pertinent to the queues. One is also able to run MPI jobs, multiple jobs and to set the environment variables required by his/her job. An option for selecting the active

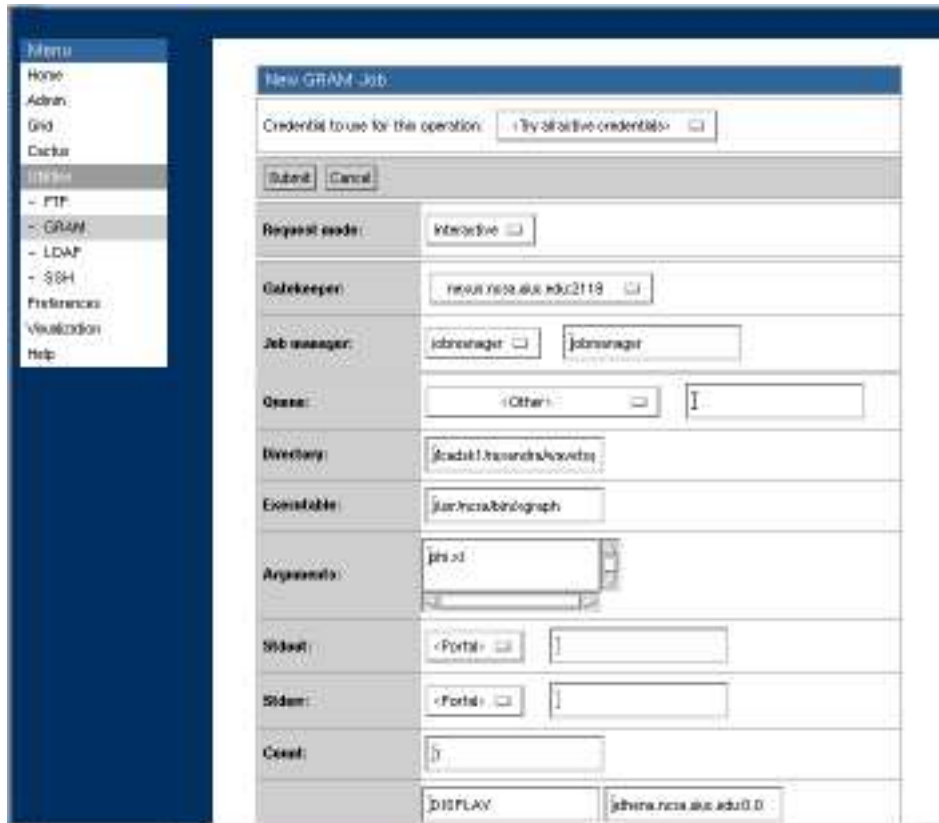


Figure 10: Submitting a GRAM Job

certificate to be used for authentication is provided. Alternatively, the portal can loop through the user's active credentials and find one that is accepted by the remote host. For interactive jobs, one can monitor the progress of a submitted job via an interactive java applet or if preferred one can also get a plain html display. The status of the job is updating continuously until completion. The user also has the ability to cancel a job that has not been completed.

For batch submissions, the user can see if his job is pending (i.e., waiting in a queue), is in progress (i.e., active), or is done. The user is able to remove pending jobs from a queue, but can not cancel a job that is already executing on a backend node.

The portal registers each job submitted into a database. This enables the user to view details of an archived gram job (the complete RSL script, the job output, the status of the job, the job environment, etc.) and to stage it back into the interface if desired.

- Jobs

We are using the GramJob, GramAttributes and GassServer Java CoG classes. The GramAttributes object represents the full RSL script and the GASS server is used to communicate the standard input, output and error of an interactive job. The Globus Resource Specification Language provides a common language to describe jobs and the resources required to run them. The ASC Portal takes the information it receives from the user and crafts an RSL

string. Afterwards, it contacts the gatekeeper on the specified machine and performs mutual authentication. The gatekeeper contacts the job manager service with your request.

Reload the gram jobs list belonging to the present portal user. It contains jobs run through the GRAM interface and through other interfaces in the portal that perform GRAM submissions (e.g., When you run a Cactus simulation, when you start a visualization program, when you perform a cvs checkout). This list shows the resource to which the job was submitted, the time stamp of the job submission, and the job status. The link beneath the request leads you to the view page.

- Gatekeepers

Uploads the collection of globus gatekeepers contained in the portal database. A detailed description can be found in the Grid->Services section of the portal.

- Globusrun

Provides a convenient interface for editing and running a RSL script from the Portal using globusrun command.

- New

To start a GRAM job just click on the “New” button from the Utilities->GRAM section of the portal. You will be presented with an thorough interface that specifies the RSL (Globus Resource Specification Language) parameters for your job:

- Hostname: The name of the machine that the portal will submit the job to. To find out what hosts are accessible from the portal, press the “Gatekeepers” option on the top part of the main menu.
- Port: The default port for Globus gatekeepers is 2119. As you can see from the Gatekeepers list, it is the same for all the listed machines.
- Resource manager: The Portal asks you to choose the job manager for the submission. The default value, “jobmanager”, is the standard interactive “fork” service. For a batch job, select the appropriate option for the Grid resource. There should be a concordance between the request mode selected and this jobmanager specification. For example, if you are running a batch job on the NCSA Origin system, request mode should be set to ”Batch” and “Resource manager” set to jobmanager-lsf.
- Queue: The batch queue into which you submit the job.
- Directory: The working directory for the simulation.
- Executable: The program you want to run. You should also specify the file path.
- Arguments: The list of command line arguments for the program.
- Stdout: The standard output for the program; A file path (absolute or relative) or URL.
- Stderr: A file path or URL to which the standard error will be written.
- Count: The number of processes to run. The default is 1.
- Environment: A list of environment variables (name/value pairs) that set the run time environment of the job.
- MaxTime: The maximum wall clock or CPU runtime in minutes (scheduler’s choice).
- MaxWallTime: The maximum wall clock runtime in minutes.

- MaxCPUTime: The maximum CPU runtime in minutes.
 - Job type : The options for Job type are “mpi”, “single”, “multiple”, and “condor”.
 - * mpi: Run the program in parallel using mpirun. The number of threads is Count.
 - * single: Run a single instance of the program, and let the program start the other Count-1 processes.
 - * multiple: Start Count instances of the program using the appropriate scheduler mechanism.
 - * condor: Start Count Condor process running in the ”standard universe”.
 - Gram My Job: Defines how the Globus GramMyjob library will operate on the Count processes. This means that if you select the “collective” option it will treat all Count processes as a part of a single job. If you select the “independent” option it will treat each of the Count processes as an independent one processor job.
 - Project: the account against which to charge.
 - HostCount: On SMP multi-computers it represents the number of nodes to distribute the Count processes across.
 - Dry Run: If you set the Dry Run on “yes”, this means that you do not want to actually run the job.
 - the Maximum Memory is the Maximum amount of memory you want to have for each process in megabytes.
 - the Minimum Memory is the Minimum amount of memory needed for each process in megabytes.
- Stage

This feature is a great time saver when running identical or slightly modified jobs that require elaborate setup (e.g., batch submission with many fields and environment variables). Selecting a job from the GRAM submissions list and pressing the Stage button will present the GRAM submission interface with fields populated by the values of the selected job.
 - View

Displays detailed information about the selected GRAM job including the RSL script and standard output for interactive jobs. The view interface also enables you to cancel a submitted GRAM job. You can cancel an interactive job that has not been completed. For batch submissions, you are able to remove pending jobs from a queue, but can not cancel a job that is already executing on a backend node.
 - Delete

Deletes the selected gram request(s) from the portal database. Note that deleting the job request does not cancel the job on the grid resource; it only affects the portal records.

4.3 LDAP

LDAP is a TCP/IP based protocol used to control communication between a user and a directory. A directory is a specialized database that is designed to retrieve information quickly and securely.

It is optimized for read access because the type of information in the directory is searched often, but changes infrequently. The information in the directory is organized in a simple tree structure denoted as the Directory Information Tree (DIT).

The interface of the “LDAP” section of the “Utilities” menu enables the user to obtain detailed information on the machines, computers and networks available on the Grid. After clicking on the “LDAP” option of the Main Menu you will see a list of available LDAP servers, the most important of which is the central organizational GIIS (Grid Index Information Service) server of the ASC virtual machine room, giis.ascportal.org. Also included in the list are the GRIS (Grid Resource Information Service) servers that run on the individual resources and report their contents to the central GIIS. On the Upper Menu you will see two options “Browse” and “Query”. The Metacomputing Directory Service (MDS), also known as Grid Information Service (GIS), is a service that allows the storage of information about the state of the Grid infrastructure. It publishes information via the LDAP protocol. The Grid Information Service has the ability to retrieve information associated with a particular name. You use this feature in the “Browse” section of the “LDAP” option of ASC Portal. The GIS can also retrieve a list of categorized entities. Such categories are define by “object classes”. A powerful extension to this functionality of GIS is the ability to augment the lookups with Boolean search filters. You are using this functionality when you perform a “Query”.

The implementation that presently exists within GridSphere operates with the MDS corresponding to Globus 1.1.x. Upgrading of the tools to accommodate the full functionality of MDS 2.1 (compatible with Globus 2.0) is necessary. Some anonymous browsing of a server is possible. For example the new Alliance GIIS on ll01.alliance.unm.edu with Base DN of ”`mds-vo-name=Alliance,o=grid`” is available.

5 Preferences

A brief interface for configuring a user profile with simple personal information is present in this section. We anticipate that this section will be enhanced to map collections of resources and services to a user’s profile within the portal. Upcoming additions will let the user set up email notification for specific events that occur with the portal (e.g., simulation initiation and completion, location of output data files, etc)

The Visualization section provides a a general framework for starting up applications for the graphical display of data sets. By pressing a button one can launch any visualization program present on the remote host. The user accomplishes this by constructing the command line to be executed on on the remote machine.

6 Visualization

To help the user build the command line the interface contains a file listing interface and a text area of visualization notes. The file listing interface displays any readable directories or files on the

remote machine. The user can find present visualization programs and data files. A full Unix file description (complete file name, file size, date last modified) is displayed when the file/directory is selected. The file listing can be ordered by the filename extensions. One is also able to keep “Visualization Notes” in a text area on the visualization page. The default visualization notes contain a list with the location of visualization tools (Vision2.0, xgraph, gnuplot, etc.) on the various supercomputers. This interface assumes an X environment for the remote display.

7 Conclusion

Please log onto the portal and try out its numerous features! Good Luck!